Exercitiul 3

De ce in algoritmul Bakery (discutat in cursul 2, al carui pseudocod e amintit mai jos), în comparatia tuplelor din metoda lock (label[i],i) > (label[k],k) nu este utilizata doar comparatia etichetei (label) sau doar a indexului ce identifica threadul, ci e necesara comparatia tuplelor formate din ambele? Argumentați răspunsul descriind o situatie concreta pentru doua thread-uri care ar folosi doar etichetele sau indecsii în comparatia respectivă.

```
class Bakery implements Lock {
  boolean[] flag;
  Label[] label;

  public Bakery (int n) {
        flag  = new boolean[n];
        label = new Label[n];
        for (int i = 0; i < n; i++) {
                flag[i] = false; label[i] = 0;
        }
  }

  public void lock() {
        flag[i]  = true;
        label[i] = max(label[0], ... ,label[n-1])+1;
        while (exists k!=i with flag[k]==true && (label[i],i) > (label[k],k)) {};
  }

  public void unlock() {
        flag[i] = false;
  }
}
```

In algoritmul bakery, rolul etichetei este de a memora o ordine în care firele au intrat în secțiunea critică, fiecare fir are o valoare ce reprezinta a cata intrare in sectiunea critica se realizeaza, iar rolul indexului este de a rezolva potențialele conflicte cauzate de valori identice ale etichetei.

- **Cazul I** (comparatia se realizeaza folosind doar eticheta):
  În cazul în care singurul criteriu de verificare al firelor pentru intrarea în secțiunea citica este valoare etichetei, exista posibilitatea de **blocaj** atunci cand doua fire au aceasi valoare în urma unei atribuiri simultane de eticheta:
    1. T1 intra in lock in acelasi timp ca si T2
    2. T1 si T2 isi seteaza flag la true
    3. T1 și T2 își setează label în același timp (valoarea 3 ca exemplu)
    4. Ambele fire intra in while, T1 este reprezentat de indexul i și T2 este k unde k != i, ambele flag sunt true, deci se trece la comparatie
    5. Algoritmul (din firul T1) așteaptă terminarea execuției tuturor firelor cu eticheta mai mica decat 3, dar îl întâmpina pe T2 cu eticheta cu aceeași valoare, deci continua sa astepte terminarea lui T2 fiindcă label[i] <= label[k]
    6. Algoritmul (din firul T2) intra în aceasi situatie, așteaptă terminarea lui T1 label[i] <= label[k]
    7. Programul ramane suspendat in blocaj

- **Cazul II (**comparatia se realizeaza folosind doar indexul):
  În acest caz pot apărea **probleme de corectitudine** fiindcă programul nu mai garantează respectarea ordinii de intrare în secțiunea critică fiindcă acestea nu mai sunt luate în considerare:
    1. Firul T2 intra in lock, flag[2] = true si label[2] = 1
    2. Firul T1 intra in lock, flag[1] = true si label[1] = 2
    3. Ambele fire intra in while, T2 este reprezentat de indexul i și T1 este k unde k != i, ambele flag sunt true, deci se trece la comparatie
    4. Se compara indecsi (în algoritmul din firul T2), indexul lui T2 este mai mare decat indexul lui T1, deci firul T2 trebuie sa aștepte finalizarea execuției firului T1 în ciuda faptului că valoarea din label ar indica faptul ca ar urma T2 sa intra în secțiunea critică
    5. T1 intra in sectiunea critica, își setează flag-ul pe false si i se permite lui T2 sa intre
    6. În acest moment rezultatul după execuția celor doua fire n-are garanția de a fi corect din cauza nerespectării ordinii de intrare în secțiunea critică

In concluzie, comparatia tuplelor formate din ericheta si index este necesara pentrua prevenii blocaje prin rezolvarea conflictelor de valori identice ale etichetei, și pentru a asigura corectitudinea programului prin urmărirea și respectarea ordinii de intrare ale firelor în secțiunea critică.